Latent Embedding Feedback and Discriminative Features for Zero-Shot Classification

Narayan, S.; Gupta, A.; Khan, F.S.; Snoek, C.G.M.; Shao, L.

DOI
10.1007/978-3-030-58542-6_29

Publication date
2020

Document Version
Final published version

Published in
Computer Vision – ECCV 2020

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Citation for published version (APA):
Latent Embedding Feedback
and Discriminative Features for Zero-Shot Classification

Sanath Narayan\textsuperscript{1(\$)}, Akshita Gupta\textsuperscript{1}, Fahad Shahbaz Khan\textsuperscript{1,3},
Cees G. M. Snoek\textsuperscript{2}, and Ling Shao\textsuperscript{1,3}

\textsuperscript{1} Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
sanath.narayan@inceptioniai.org
\textsuperscript{2} University of Amsterdam, Amsterdam, The Netherlands
\textsuperscript{3} Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

Abstract. Zero-shot learning strives to classify unseen categories for which no data is available during training. In the generalized variant, the test samples can further belong to seen or unseen categories. The state-of-the-art relies on Generative Adversarial Networks that synthesize unseen class features by leveraging class-specific semantic embeddings. During training, they generate semantically consistent features, but discard this constraint during feature synthesis and classification. We propose to enforce semantic consistency at all stages of (generalized) zero-shot learning: training, feature synthesis and classification. We first introduce a feedback loop, from a semantic embedding decoder, that iteratively refines the generated features during both the training and feature synthesis stages. The synthesized features together with their corresponding latent embeddings from the decoder are then transformed into discriminative features and utilized during classification to reduce ambiguities among categories. Experiments on (generalized) zero-shot object and action classification reveal the benefit of semantic consistency and iterative feedback, outperforming existing methods on six zero-shot learning benchmarks. Source code at https://github.com/akshitac8/tfvaegan.

Keywords: Generalized zero-shot classification · Feature synthesis

1 Introduction

This paper strives for zero-shot learning, a challenging vision problem that involves classifying images or videos into new (“unseen”) categories at test time, without having been provided any corresponding visual example during training.

S. Narayan and A. Gupta—Equal Contribution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58542-6_29) contains supplementary material, which is available to authorized users.

© Springer Nature Switzerland AG 2020
https://doi.org/10.1007/978-3-030-58542-6_29
In the literature \cite{1,34,42,45}, this is typically achieved by utilizing the labelled seen class instances and class-specific semantic embeddings (provided as a side information), which encode the inter-class relationships. Different from the zero-shot setting, the test samples can belong to the seen or unseen categories in generalized zero-shot learning \cite{41}. In this work, we investigate the problem of both zero-shot learning (ZSL) and generalized zero-shot learning (GZSL).

Most recent ZSL and GZSL recognition approaches \cite{8,13,22,42,43} are based on Generative Adversarial Networks (GANs) \cite{11}, which aim at directly optimizing the divergence between real and generated data. The work of \cite{42} learns a GAN using the seen class feature instances and the corresponding class-specific semantic embeddings, which are either manually annotated or word vector \cite{27} representations. Feature instances of the unseen categories, whose real features are unavailable during training, are then synthesized using the trained GAN and used together with the real feature instances from the seen categories to train zero-shot classifiers in a fully-supervised setting. A few works \cite{8,13,25} additionally utilize auxiliary modules, such as a decoder, to enforce a cycle-consistency constraint on the reconstruction of semantic embeddings during training. Such an auxiliary decoder module aids the generator to synthesize semantically consistent features. Surprisingly, these modules are only employed during training and discarded during both the feature synthesis and ZSL classification stages. Since the auxiliary module aids the generator during training, it is also expected to help obtain discriminative features during feature synthesis and reduce the ambiguities among different classes during classification. In this work, we address the issues of enhanced feature synthesis and improved zero-shot classification.

Further, GANs are likely to encounter mode collapse issues \cite{2}, resulting in decreased diversity of generated features. While Variational Autoencoders (VAEs) \cite{18} achieve more stable feature generation, the approximate inference distribution is likely to be different from the true posterior \cite{48}. Recently, \cite{43} build on \cite{42} to combine the strengths of VAEs and GANs and introduce an \textit{f-VAEGAN} ZSL framework by sharing the VAE decoder and GAN generator modules. To ensure that the generated features are semantically close to the distribution of real feature, a cycle-consistency loss \cite{49} is employed between generated and original features, during training. Here, we propose to additionally enforce a similar consistency loss on the semantic embeddings during training and further utilize the learned information during feature synthesis and classification.

\subsection{1.1 Contributions}

We propose a novel method, which advocates the effective utilization of a semantic embedding decoder (SED) module at all stages of the ZSL framework: training, feature synthesis and classification. Our method is built on a VAE-GAN architecture. (i) We design a feedback module for (generalized) zero-shot learning that utilizes SED during both training and feature synthesis stages. The feedback module first transforms the latent embeddings of SED, which are then used to modulate the latent representations of the generator. To the best of
our knowledge, we are the first to propose a feedback module, within a VAE-GAN architecture, for the problem of (generalized) zero-shot recognition. (ii) We introduce a discriminative feature transformation, during the classification stage, that utilizes the latent embeddings of SED along with their corresponding visual features for reducing ambiguities among object categories. In addition to object recognition, we show effectiveness of the proposed approach for (generalized) zero-shot action recognition in videos.

We validate our approach by performing comprehensive experiments on four commonly used ZSL object recognition datasets: CUB [40], FLO [29], SUN [30] and AWA [41]. Our experimental evaluation shows the benefits of utilizing SED at all stages of the ZSL/GZSL pipeline. In comparison to the baseline, the proposed approach obtains absolute gains of 4.6%, 7.1%, 1.7%, and 3.1% on CUB, FLO, SUN, and AWA, respectively for generalized zero-shot (GZSL) object recognition. In addition to object recognition, we evaluate our method on two (generalized) zero-shot action recognition in videos datasets: HMDB51 [20] and UCF101 [38]. Our approach outperforms existing methods on all six datasets. We also show the generalizability of our proposed contributions by integrating them into GAN-based (generalized) zero-shot recognition framework.

2 Related Work

In recent years, the problem of object recognition under zero-shot learning (ZSL) settings has been well studied [1,9,10,16,33,34,42,45]. Earlier ZSL image classification works [16,21] learn semantic embedding classifiers for associating seen and unseen classes. Different from these methods, the works of [1,9,34] learn a compatibility function between the semantic embedding and visual feature spaces. Other than these inductive approaches that rely only on the labelled data from seen classes, the works of [10,33,45] leverage additional unlabelled data from unseen classes through label propagation under a transductive zero-shot setting.

Recently, Generative Adversarial Networks [11] (GANs) have been employed to synthesize unseen class features, which are then used in a fully supervised setting to train ZSL classifiers [8,22,42,43]. A conditional Wasserstein GAN [3] (WGAN) is used along with a seen category classifier to learn the generator for unseen class feature synthesis [42]. This is achieved by using a WGAN loss and a classification loss. In [8], the seen category classifier is replaced by a decoder together with the integration of a cycle-consistency loss [49]. The work of [35] proposes an approach where cross and distribution alignment losses are introduced for aligning the visual features and corresponding embeddings in a shared latent space, using two Variational Autoencoders [18] (VAEs). The work of [43] introduces a f-VAEGAN framework which combines a VAE and a GAN by sharing the decoder of VAE and generator of GAN for feature synthesis. For training, the f-VAEGAN framework utilizes a cycle-consistency constraint between generated and original visual features. However, a similar constraint is not enforced on the semantic embeddings in their framework. Different from f-VAEGAN, other
GAN-based ZSL classification methods [8,13,25,47] investigate the utilization of auxiliary modules to enforce cycle-consistency on the embeddings. Nevertheless, these modules are utilized only during training and discarded during both feature synthesis and ZSL classification stages.

Previous works [14,23,36,46] have investigated leveraging feedback information to incrementally improve the performance of different applications, including classification, image-to-image translation and super-resolution. To the best of our knowledge, our approach is the first to incorporate a feedback loop for improved feature synthesis in the context of (generalized) zero-shot recognition (both image and video). We systematically design a feedback module, in a VAE-GAN framework, that iteratively refines the synthesized features for ZSL.

While zero-shot image classification has been extensively studied, zero-shot action recognition in videos received less attention. Several works [19,28,44] study the problem of zero-shot action recognition in videos under transductive setting. The use of image classifiers and object detectors for action recognition under ZSL setting are investigated in [15,26]. Recently, GANs have been utilized to synthesize unseen class video features in [25,47]. Here, we further investigate the effectiveness of our framework for zero-shot action recognition in videos.

3 Method

We present an approach, TF-VAEGAN, for (generalized) zero-shot recognition. As discussed earlier, the objective in ZSL is to classify images or videos into new classes, which are unknown during the training stage. Different from ZSL, test samples can belong to seen or unseen classes in the GZSL setting, thereby making it a harder problem due to the domain shift between the seen and unseen classes. Let \( x \in \mathcal{X} \) denote the encoded feature instances of images (videos) and \( y \in \mathcal{Y}^s \) the corresponding labels from the set of \( M \) seen class labels \( \mathcal{Y}^s = \{y_1, \ldots, y_M\} \). Let \( \mathcal{Y}^u = \{u_1, \ldots, u_N\} \) denote the set of \( N \) unseen classes, which is disjoint from the seen class set \( \mathcal{Y}^s \). The seen and unseen classes are described by the category-specific semantic embeddings \( a(k) \in \mathcal{A}, \forall k \in \mathcal{Y}^s \cup \mathcal{Y}^u \), which encode the relationships among all the classes. While the unlabelled test features \( x_t \in \mathcal{X} \) are not used during training in the inductive setting, they are used during training in the transductive setting to reduce the bias towards seen classes. The tasks in ZSL and GZSL are to learn the classifiers \( f_{zsl} : \mathcal{X} \to \mathcal{Y}^u \) and \( f_{gzsl} : \mathcal{X} \to \mathcal{Y}^s \cup \mathcal{Y}^u \), respectively. To this end, we first learn to synthesize the features using the seen class features \( x_s \) and corresponding embeddings \( a(y) \).

The learned model is then used to synthesize unseen class features \( \hat{x}_u \) using the unseen class embeddings \( a(u) \). The resulting synthesized features \( \hat{x}_u \), along with the real seen class features \( x_s \), are further deployed to train the final classifiers \( f_{zsl} \) and \( f_{gzsl} \).

3.1 Preliminaries: f-VAEGAN

We base our approach on the recently introduced f-VAEGAN [43], which combines the strengths of the VAE [18] and GAN [11] as discussed earlier, achieving
impressive results for ZSL classification. Compared to GAN based models, e.g., f-CLS WGAN [42], the f-VAEGAN [43] generates semantically consistent features by sharing the decoder and generator of the VAE and GAN. In f-VAEGAN, the feature generating VAE [18] (f-VAE) comprises an encoder $E(x, a)$, which encodes an input feature $x$ to a latent code $z$, and a decoder $G(z, a)$ (shared with f-WGAN, as a conditional generator) that reconstructs $x$ from $z$. Both $E$ and $G$ are conditioned on the embedding $a$, optimizing,

$$\mathcal{L}_V = \text{KL}(E(x, a)||p(z|a)) - \mathbb{E}_{E(x,a)}[\log G(z, a)],$$  \tag{1}\]

where $\text{KL}$ is the Kullback-Leibler divergence, $p(z|a)$ is a prior distribution, assumed to be $\mathcal{N}(0, 1)$ and $\log G(z, a)$ is the reconstruction loss. The feature generating network [42] (f-WGAN) comprises a generator $G(z, a)$ and a discriminator $D(x, a)$. The generator $G(z, a)$ synthesizes a feature $\hat{x} \in \mathcal{X}$ from a random input noise $z$, whereas the discriminator $D(x, a)$ takes an input feature $x$ and outputs a real value indicating the degree of realness or fakeness of the input features. Both $G$ and $D$ are conditioned on the embedding $a$, optimizing the WGAN loss $\mathcal{L}_W = \mathbb{E}[D(x, a)] - \mathbb{E}[D(\hat{x}, a)] - \lambda \mathbb{E}[||\nabla D(\hat{x}, a)||_2^2]$. Here, $\hat{x} = G(z, a)$ is the synthesized feature, $\lambda$ is the penalty coefficient and $\hat{x}$ is sampled randomly from the line connecting $x$ and $\hat{x}$. The f-VAEGAN is then optimized by:

$$\mathcal{L}_{vaegan} = \mathcal{L}_V + \alpha \mathcal{L}_W,$$  \tag{2}\]

where $\alpha$ is a hyper-parameter. For more details, we refer to [43].

**Limitations:** The loss formulation for training f-VAEGAN, contains a constraint (second term in Eq. 1) that ensures the generated visual features are cyclically-consistent, at train time, with the original visual features. However, a similar cycle-consistency constraint is not enforced on the semantic embeddings. Alternatively, other GAN-based ZSL methods [8,47] utilize auxiliary modules (apart from the generator) for achieving cyclic-consistency on embeddings. However, these modules are employed only during training and discarded at both feature synthesis and ZSL classification stages. In this work, we introduce a semantic embedding decoder (SED) that enforces cycle-consistency on semantic embeddings and utilize it at all stages: training, feature synthesis and ZSL classification. We argue that the generator and SED contain complementary information with respect to feature instances, since the two modules perform inverse transformations in relation to each other. The generator module transforms the semantic embeddings to the feature instances whereas, SED transforms the feature instances to semantic embeddings. Our approach focuses on the utilization of this complementary information for improving feature synthesis and reducing ambiguities among classes (e.g., fine-grained classes) during ZSL classification.

### 3.2 Overall Architecture

The overall architecture is illustrated in Fig. 1. The VAE-GAN consists of an encoder $E$, generator $G$ and discriminator $D$. The input to $E$ are the real features
Fig. 1. **Proposed architecture** (Sect. 3.2). Given a seen class image, visual features $x$ are extracted from the backbone network and input to the encoder $E$, along with the corresponding semantic embeddings $a$. The encoder $E$ outputs a latent code $z$, which is then input together with embeddings $a$ to the generator $G$ that synthesizes features $\hat{x}$. The discriminator $D$ learns to distinguish between real and synthesized features $x$ and $\hat{x}$, respectively. Both $E$ and $G$ together constitute the VAE, which is trained using a binary cross-entropy loss ($L_{BCE}$) and the KL divergence ($L_{KL}$). Similarly, both $G$ and $D$ form the GAN trained using the WGAN loss ($L_{W}$). A semantic embedding decoder $\text{Dec}$ is introduced (Sect. 3.3) to reconstruct the embeddings $\hat{a}$ using a cycle-consistency loss ($L_{R}$). Further, a feedback module $F$ (Sect. 3.4) is integrated to transform the latent embedding $\hat{h}$ of $\text{Dec}$ and feed it back to $G$, which iteratively refines $\hat{x}$.

of seen classes $x$ and the semantic embeddings $a$ and the output of $E$ are the parameters of a noise distribution. These parameters are matched to those of a zero-mean unit-variance Gaussian prior distribution using the KL divergence ($L_{KL}$). The noise $z$ and embeddings $a$ are input to $G$, which synthesizes the features $\hat{x}$. The synthesized features $\hat{x}$ and original features $x$ are compared using a binary cross-entropy loss $L_{BCE}$. The discriminator $D$ takes either $x$ or $\hat{x}$ along with embeddings $a$ as input, and computes a real number that determines whether the input is real or fake. The WGAN loss $L_{W}$ is applied at the output of $D$ to learn to distinguish between the real and fake features.

The focus of our design is the integration of an additional semantic embedding decoder (SED) $\text{Dec}$ at both the feature synthesis and ZSL/GZSL classification stages. Additionally, we introduce a feedback module $F$, which is utilized during training and feature synthesis, along with $\text{Dec}$. Both the semantic embedding decoder $\text{Dec}$ and feedback module $F$ collectively address the objectives of enhanced feature synthesis and reduced ambiguities among categories during classification. The $\text{Dec}$ takes either $x$ or $\hat{x}$ and reconstructs the embeddings $\hat{a}$. It is trained using a cycle-consistency loss $L_{R}$. The learned $\text{Dec}$ is subsequently used in the ZSL/GZSL classifiers. The feedback module $F$ transforms the latent embedding of $\text{Dec}$ and feeds it back to the latent representation of generator $G$ in order to achieve improved feature synthesis. The SED $\text{Dec}$ and feedback module $F$ are described in detail in Sect. 3.3 and 3.4.
3.3 Semantic Embedding Decoder

Here, we introduce a semantic embedding decoder $Dec : \mathcal{X} \rightarrow \mathcal{A}$, for reconstructing the semantic embeddings $a$ from the generated features $\hat{x}$. Enforcing a cycle-consistency on the reconstructed semantic embeddings ensures that the generated features are transformed to the same embeddings that generated them. As a result, semantically consistent features are obtained during feature synthesis. The cycle-consistency of the semantic embeddings is achieved using the $\ell_1$ reconstruction loss as follows:

$$L_R = \mathbb{E}[||Dec(x) - a||_1] + \mathbb{E}[||Dec(\hat{x}) - a||_1].$$  \hspace{1cm} (3)

The loss formulation for training the proposed TF-VAEGAN is then given by,

$$L_{total} = L_{vaegan} + \beta L_R,$$

where $\beta$ is a hyper-parameter for weighting the decoder reconstruction error.

As discussed earlier, existing GAN-based ZSL approaches [8,47] employ a semantic embedding decoder (SED) only during training and discard it during both unseen class feature synthesis and ZSL classification stage. In our approach, SED is utilized at all three stages of VAE-GAN based ZSL pipeline: training, feature synthesis and classification. Next, we describe importance of SED during classification and later investigate its role during feature synthesis (Sect. 3.4).

**Discriminative Feature Transformation:** Here, we describe the proposed discriminative feature transformation scheme to effectively utilize the auxiliary information in semantic embedding decoder (SED) at the ZSL classification stage. The generator $G$ learns a per-class “single semantic embedding to many instances” mapping using only the seen class features and embeddings. Similar to the generator $G$, the SED is also trained using only the seen classes but learns a per-class “many instances to one embedding” inverse mapping. Thus, the generator $G$ and SED $Dec$ are likely to encode complementary information of the categories. Here, we propose to use the latent embedding from SED as a useful source of information at the classification stage (see Fig. 2a) for reducing ambiguities among features instances of different categories.

First, the training of feature generator $G$ and semantic embedding decoder $Dec$ is performed. Then, $Dec$ is used to transform the features (real and synthesized) to the embedding space $\mathcal{A}$. Afterwards, the latent embeddings from $Dec$ are concatenated with the respective visual features. Let $h_s$ and $\hat{h}_u \in \mathcal{H}$ denote the hidden layer (latent) embedding from the $Dec$ for inputs $x_s$ and $\hat{x}_u$, respectively. The transformed features are represented by: $x_s \oplus h_s$ and $\hat{x}_u \oplus \hat{h}_u$, where $\oplus$ denotes concatenation. In our method, the transformed features are used to learn final ZSL and GZSL classifiers as,

$$f_{zsl} : \mathcal{X} \oplus \mathcal{H} \rightarrow \mathcal{Y}^u \quad \text{and} \quad f_{gzsl} : \mathcal{X} \oplus \mathcal{H} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u.$$  \hspace{1cm} (5)

As a result, the final classifiers learn to better distinguish categories using transformed features. Next, we describe integration of $Dec$ during feature synthesis.
Fig. 2. (a) Integration of semantic embedding decoder Dec at the ZSL/GZSL classification stage. A feature transformation is performed by concatenating (⊕) the input visual features $x$ with the corresponding latent embedding $h$ from SED. The transformed discriminative features are then used for ZSL/GZSL classification. (b) Feedback module overview. First sub-iteration: The generator $G$ synthesizes initial features $\hat{x}[0]$ using the noise $z$ and embeddings $a$. The initial features are passed through the Dec. Second sub-iteration: The module $F$ transforms the latent embedding $h$ from Dec to $\hat{x}^{f}$, which represents the feedback to $G$. The generator $G$ synthesizes enhanced features $\hat{x}[1]$ using the same $z$ and $a$ along with the feedback $\hat{x}^{f}$.

3.4 Feedback Module

The baseline f-VAEGAN does not enforce cycle-consistency in the attribute space and directly synthesizes visual features $\hat{x}$ from the class-specific embeddings $a$ via the generator (see Fig. 3a). This results in a semantic gap between the real and synthesized visual features. To address this issue, we introduce a feedback loop that iteratively refines the feature generation (see Fig. 3b) during both the training and synthesis stages. The feedback loop is introduced from the semantic embedding decoder Dec to the generator $G$, through our feedback module $F$ (see Fig. 1 and Fig. 2b). The proposed module $F$ enables the effective utilization of Dec during both training and feature synthesis stages. Let $g^{l}$ denote the $l^{th}$ layer output of $G$ and $\hat{x}^{f}$ denote the feedback component that additively modulates $g^{l}$. The feedback modulation of output $g^{l}$ is given by,

$$g^{l} \leftarrow g^{l} + \delta \hat{x}^{f},$$

where $\hat{x}^{f} = F(h)$, with $h$ as the latent embedding of Dec and $\delta$ controls the feedback modulation. To the best of our knowledge, we are the first to design and incorporate a feedback loop for zero-shot recognition. Our feedback loop is based on [36], originally introduced for image super-resolution. However, we observe that it provides sub-optimal performance for zero-shot recognition due to its less reliable feedback during unseen class feature synthesis. Next, we describe an improved feedback loop with necessary modifications for zero-shot recognition.

Feedback Module Input: The adversarial feedback employs a latent representation of an unconditional discriminator $D$ as its input [36]. However, in the ZSL problem, $D$ is conditional and is trained with an objective to distinguish between the real and fake features of the seen categories. This restricts $D$ from providing reliable feedback during unseen class feature synthesis. In order to
overcome this limitation, we turn our attention to semantic embedding decoder \( \text{Dec} \), whose aim is to reconstruct the class-specific semantic embeddings from features instances. Since \( \text{Dec} \) learns class-specific transformations from visual features to the semantic embeddings, it is better suited (than \( \text{D} \)) to provide feedback to generator \( \text{G} \).

**Training Strategy:** Originally, the feedback module \( \text{F} \) is trained in a two-stage fashion [36], where the generator \( \text{G} \) and discriminator \( \text{D} \) are first fully trained, as in the standard GAN training approach. Then, \( \text{F} \) is trained using a feedback from \( \text{D} \) and freezing \( \text{G} \). Since, the output of \( \text{G} \) improves due to the feedback from \( \text{F} \), the discriminator \( \text{D} \) is continued to be trained alongside \( \text{F} \), in an adversarial manner. In this work, we argue that such a two-stage training strategy is sub-optimal for ZSL, since \( \text{G} \) is always fixed and not allowed to improve its feature synthesis. To further utilize the feedback for improved feature synthesis, \( \text{G} \) and \( \text{F} \) are trained alternately in our method. In our alternating training strategy, the generator training iteration is unchanged. However, during the training iterations of \( \text{F} \), we perform two sub-iterations (see Fig. 2b).

**First Sub-iteration:** The noise \( z \) and semantic embeddings \( a \) are input to the generator \( \text{G} \) to yield an initial synthesized feature \( \hat{x}[0] = G(z, a) \), which is then passed through to the semantic embedding decoder \( \text{Dec} \).

**Second Sub-iteration:** The latent embedding \( \hat{h} \) from \( \text{Dec} \) is input to \( \text{F} \), resulting in an output \( \hat{x}^f[t] = F(\hat{h}) \), which is added to the latent representation (denoted as \( g^l \) in Eq. 6) of \( \text{G} \). The same \( z \) and \( a \) (used in the first sub-iteration) are used as input to \( \text{G} \) for the second sub-iteration, with the additional input \( \hat{x}^f[t] \) added to the latent representation \( g^l \) of generator \( \text{G} \). The generator then outputs a synthesized feature \( \hat{x}[t + 1] \), as,

\[
\hat{x}[t + 1] = G(z, a, \hat{x}^f[t]). \tag{7}
\]

The refined feature \( \hat{x}[t + 1] \) is input to \( \text{D} \) and \( \text{Dec} \), and corresponding losses are computed (Eq. 4) for training. In practice, the second sub-iteration is performed only once. The feedback module \( \text{F} \) allows generator \( \text{G} \) to view the latent
embedding of $Dec$, corresponding to current generated features. This enables $G$ to appropriately refine its output (feature generation) iteratively, leading to an enhanced feature representation.

3.5 (Generalized) Zero-Shot Classification

In our TF-VAEGAN, unseen class features are synthesized by inputting respective embeddings $a(u)$ and noise $z$ to $G$, given by $\hat{x}_u = G(z, a(u), \hat{x}^f[0])$. Here, $\hat{x}^f[0]$ denotes feedback output of $F$, computed for the same $a(u)$ and $z$. The synthesized unseen class features $\hat{x}_u$ and real seen class features $x_s$ are further input to $Dec$ to obtain their respective latent embeddings, which are concatenated with input features. In this way, we obtain transformed features $x_s \oplus h_s$ and $\hat{x}_u \oplus \hat{h}_u$, which are used to train ZSL and GZSL classifiers, $f_{zsl}$ and $f_{gzsl}$, respectively. At inference, test features $x_t$ are transformed in a similar manner, to obtain $x_t \oplus h_t$. The transformed features are then input to classifiers for final predictions.

4 Experiments

Datasets: We evaluate our TF-VAEGAN framework on four standard zero-shot object recognition datasets: Caltech-UCSD-Birds [40] (CUB), Oxford Flowers [29] (FLO), SUN Attribute [30] (SUN), and Animals with Attributes2 [41] (AWA2) containing 200, 102, 717 and 50 categories, respectively. For fair comparison, we use the same splits, evaluation protocols and class embeddings as in [41].

Visual Features and Embeddings: We extract the average-pooled feature instances of size 2048 from the ImageNet-1K [6] pre-trained ResNet-101 [12]. For semantic embeddings, we use the class-level attributes for CUB (312-d), SUN (102-d) and AWA2 (85-d). For FLO, fine-grained visual descriptions of image are used to extract 1024-d embeddings from a character-based CNN-RNN [32].

Implementation Details: The discriminator $D$, encoder $E$ and generator $G$ are implemented as two-layer fully-connected (FC) networks with 4096 hidden units. The dimensions of $z$ and $a$ are set to be equal ($R^{d_z} = R^{d_a}$). The semantic embedding decoder $Dec$ and feedback module $F$ are also two-layer FC networks with 4096 hidden units. The input and output dimensions of $F$ are set to 4096 to match the hidden units of $Dec$ and $G$. For transductive setting, an unconditional discriminator $D2$ is employed for utilizing the unlabelled feature instances during training, as in [43]. Since the corresponding semantic embeddings are not available for unlabelled instances, only the visual feature is input to $D2$. Leaky ReLU activation is used everywhere, except at the output of $G$, where a sigmoid activation is used for applying BCE loss. The network is trained using the Adam optimizer with $10^{-4}$ learning rate. Final ZSL/GZSL classifiers are single layer FC networks with output units equal to number of test classes. Hyper-parameters $\alpha$, $\beta$ and $\delta$ are set to 10, 0.01 and 1, respectively. The gradient penalty coefficient $\lambda$ is initialized to 10 and WGAN is trained, similar to [3].
4.1 State-of-the-Art Comparison

Table 1 shows state-of-the-art comparison on four object recognition datasets. Results for inductive (IN) and transductive (TR) settings are obtained without any fine-tuning of the backbone network. For inductive (IN) ZSL, the Cycle-WGAN [8] obtains classification scores of 58.6%, 70.3%, 59.9%, and 66.8% on CUB, FLO, SUN and AWA, respectively. The f-VAEGAN [43] reports classification accuracies of 61%, 67.7%, 64.7%, and 71.1% on the same datasets. Our TF-VAEGAN outperforms f-VAEGAN on all datasets achieving classification scores of 64.9%, 70.8%, 66.0%, and 72.2% on CUB, FLO, SUN and AWA, respectively. In the transductive (TR) ZSL setting, f-VAEGAN obtains top-1 classification (T1) accuracies of 71.1%, 89.1%, 70.1%, and 92.1% on the four datasets. Our TF-VAEGAN also performs favourably compared to existing methods on all datasets for both inductive and transductive GZSL settings. Utilizing unlabelled instances during training, to reduce the domain shift problem for unseen classes, in the transductive setting yields higher results compared to inductive setting.

Some previous works, including f-VAEGAN [43] have reported results with fine-tuning the backbone network only using the seen classes (without violating...
Table 2. Baseline performance comparison on CUB [40]. In both inductive and transductive settings, our Feedback and T-feature provide consistent improvements over the baseline for both ZSL and GZSL. Further, our final TF-VAEGAN framework, integrating both Feedback and T-feature, achieves further gains over the baseline in both inductive and transductive settings, for ZSL and GZSL.

<table>
<thead>
<tr>
<th></th>
<th>INDUCTIVE</th>
<th></th>
<th>TRANSDUCTIVE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
<td>Feedback</td>
<td>T-feature</td>
<td>TF-VAEGAN</td>
</tr>
<tr>
<td>ZSL</td>
<td>61.2</td>
<td>62.8</td>
<td>64.0</td>
<td>64.9</td>
</tr>
<tr>
<td>GZSL</td>
<td>53.5</td>
<td>54.8</td>
<td>56.9</td>
<td>58.1</td>
</tr>
</tbody>
</table>

the ZSL condition). Similarly, we also evaluate our TF-VAEGAN by utilizing fine-tuned backbone features. Table 1 shows the comparison with existing fine-tuning based methods for both ZSL and GZSL in fine-tuned inductive (FT-IN) and fine-tuned transductive (FT-TR) settings. For FT-IN ZSL, f-VAEGAN obtains classification scores of 72.9%, 70.4%, 65.6%, and 70.3% on CUB, FLO, SUN and AWA, respectively. Our TF-VAEGAN achieves consistent improvement over f-VAEGAN on all datasets, achieving classification scores of 74.3%, 74.7%, 66.7%, and 73.4% on CUB, FLO, SUN and AWA, respectively. Our approach also improves over f-VAEGAN for the FT-TR ZSL setting. In the case of FT-IN GZSL, our TF-VAEGAN achieves gains (in terms of $\mathbf{H}$) of 1.8%, 4.3%, 3.2%, and 1.5% on CUB, FLO, SUN and AWA, respectively over f-VAEGAN. A similar trend is also observed for the FT-TR GZSL setting. In summary, our TF-VAEGAN achieves promising results for various settings and backbone feature combinations.

4.2 Ablation Study

Baseline Comparison: We first compare our proposed TF-VAEGAN with the baseline f-VAEGAN [43] on CUB for (generalized) zero-shot recognition in both inductive and transductive settings. The results are reported in Table 2 in terms of average top-1 classification accuracy for ZSL and harmonic mean of the classification accuracies of seen and unseen classes for GZSL. For the baseline, we present the results based on our re-implementation. In addition to our final TF-VAEGAN, we report results of our feedback module alone (denoted as Feedback in Table 2) without feature transformation utilized at classification stage. Moreover, the performance of discriminative feature transformation alone (denoted as T-feature), without utilizing the feedback is also presented. For the inductive setting, Baseline obtains a classification performance of 61.2% and 53.5% for ZSL and GZSL. Both our contributions, Feedback and T-feature, consistently improve the performance over the baseline. The best results are obtained by our TF-VAEGAN, with gains of 3.7% and 4.6% over the baseline, for ZSL and GZSL. Similar to the inductive (IN) setting, our proposed TF-VAEGAN also achieves favourable performance in transductive (TR) setting. Figure 4 shows a comparison between baseline and our TF-VAEGAN methods, using t-SNE visualizations [24] of test instances from four example fine-grained
classes of CUB. While the baseline struggles to correctly classify these fine-grained class instances due to inter-class confusion, our TF-VAEGAN improves inter-class grouping leading to a favorable classification performance.

**Generalization Capabilities:** Here, we base our approach on a VAE-GAN architecture [43]. However, our proposed contributions (a semantic embedding decoder at all stages of the ZSL pipeline and the feedback module) are generic and can also be utilized in other GAN-based ZSL frameworks. To this end, we perform an experiment by integrating our contributions in the f-CLSWGAN [42] architecture. Figure 5 shows the comparison between the baseline f-CLSWGAN and our TF-CLSWGAN for ZSL and GZSL tasks, on all four datasets. Our TF-CLSWGAN outperforms the vanilla f-CLSWGAN in all cases for both ZSL and GZSL tasks.

**Feature Visualization:** To qualitatively assess the feature synthesis stage, we train an unconditional network to invert the feature instances back to the image space by following a similar strategy as in [7,43]. Corresponding implementation details are provided in the supplementary. The model is trained on all real feature-image pairs of the 102 classes of FLO [29]. The comparison between Baseline and our Feedback synthesized features on four example flowers is shown in Fig.6. For each flower class, a ground-truth (GT) image along with three images inverted from its GT feature, Baseline and Feedback synthesized features, respectively are shown. Generally, inverting the Feedback synthesized feature yields an image that is semantically closer to the GT image than inverting the Baseline synthesized feature. This suggests that our Feedback improves the feature synthesis stage over the Baseline, where no feedback is present.

Additional quantitative and qualitative results are given in the supplementary.
Fig. 5. Generalization capabilities. (a) ZSL and (b) GZSL performance comparison to validate the generalization capabilities of our contributions. Instead of a VAE-GAN architecture, we integrate our proposed contributions in the $\text{f-CLSWGAN}$ framework. Our $\text{TF-CLSWGAN}$ outperforms the vanilla $\text{f-CLSWGAN}$ on all datasets. Best viewed in zoom.

Fig. 6. Qualitative comparison between inverted images of Baseline synthesized features and our Feedback synthesized features on four example classes of FLO [29]. The ground-truth image and the reconstructed inversion of its real feature are also shown for each example. Our Feedback improves petal shapes (Sunflower), shape of bud and petals (Blanket flower), color (Pink primrose), black lining on petals (Balloon flower) and achieves promising improvements over Baseline. Best viewed in zoom.

5 (Generalized) Zero-Shot Action Recognition

Finally, we validate our $\text{TF-VAEGAN}$ for action recognition in videos under ZSL and GZSL. Here, we use the I3D features [5], as in the GAN-based zero-shot action classification method CEWGAN [25]. While using improved video features is likely to improve the performance of a zero-shot action recognition framework, our goal is to show that our $\text{TF-VAEGAN}$ generalizes to action classification and improves the performance using the same underlying video features. As in [25], we extract spatio-temporally pooled 4096-d I3D features from pre-trained RGB and Flow I3D networks and concatenate them to obtain 8192-d video features. Further, an out-of-distribution classifier is utilized at the classification stage, as in [25]. For HMDB51, a skip-gram model [27] is used to generate semantic embeddings of size 300, using action class names as input. For UCF101, we use semantic embeddings of size 115, provided with the dataset.

Table 3 shows state-of-the-art comparison on HMDB51 [20] and UCF101 [38]. For a fair comparison, we use the same splits, embeddings and evaluation protocols as in [25]. On HMDB51, $\text{f-VAEGAN}$ obtains classification scores of 31.1%
Table 3. State-of-the-art ZSL and GZSL comparison for action recognition. Our TF-VAEGAN performs favorably against all existing methods, on both datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>ZSL</th>
<th>GZSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMDB51</td>
<td>GGM [28]</td>
<td>20.7</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td>CLSWGAN [42]</td>
<td>20.1</td>
<td>32.7</td>
</tr>
<tr>
<td></td>
<td>CEWGAN [25]</td>
<td>20.2</td>
<td>36.1</td>
</tr>
<tr>
<td></td>
<td>Obj2Act [15]</td>
<td>24.5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ObjEmb [26]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>f-VAEGAN [43]</td>
<td>31.1</td>
<td>35.6</td>
</tr>
<tr>
<td></td>
<td>TF-VAEGAN</td>
<td>33.0</td>
<td>37.6</td>
</tr>
<tr>
<td>UCF101</td>
<td>GGM [28]</td>
<td>20.3</td>
<td>37.5</td>
</tr>
<tr>
<td></td>
<td>CLSWGAN [42]</td>
<td>17.5</td>
<td>44.4</td>
</tr>
<tr>
<td></td>
<td>CEWGAN [25]</td>
<td>20.2</td>
<td>49.4</td>
</tr>
<tr>
<td></td>
<td>Obj2Act [15]</td>
<td>38.9</td>
<td>40.4</td>
</tr>
<tr>
<td></td>
<td>ObjEmb [26]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>f-VAEGAN [43]</td>
<td>38.2</td>
<td>47.2</td>
</tr>
<tr>
<td></td>
<td>TF-VAEGAN</td>
<td>41.0</td>
<td>50.9</td>
</tr>
</tbody>
</table>

and 35.6% for ZSL and GZSL. The work of [50] provides classification results of 24.4% and 17.5% for HMDB51 and UCF101, respectively for ZSL. Note that [50] also reports results using cross-dataset training on large-scale ActivityNet [4]. On HMDB51, CEWGAN [25] obtains 30.2% and 36.1% for ZSL and GZSL. Our TF-VAEGAN achieves 33.0% and 37.6% for ZSL and GZSL. Similarly, our approach performs favourably compared to existing methods on UCF101. Hence, our TF-VAEGAN generalizes to action recognition and achieves promising results.

6 Conclusion

We propose an approach that utilizes the semantic embedding decoder (SED) at all stages (training, feature synthesis and classification) of a VAE-GAN based ZSL framework. Since SED performs inverse transformations in relation to the generator, its deployment at all stages enables exploiting complementary information with respect to feature instances. To effectively utilize SED during both training and feature synthesis, we introduce a feedback module that transforms the latent embeddings of the SED and modulates the latent representations of the generator. We further introduce a discriminative feature transformation, during the classification stage, which utilizes the latent embeddings of SED along with respective features. Experiments on six datasets clearly suggest that our approach achieves favorable performance, compared to existing methods.

References